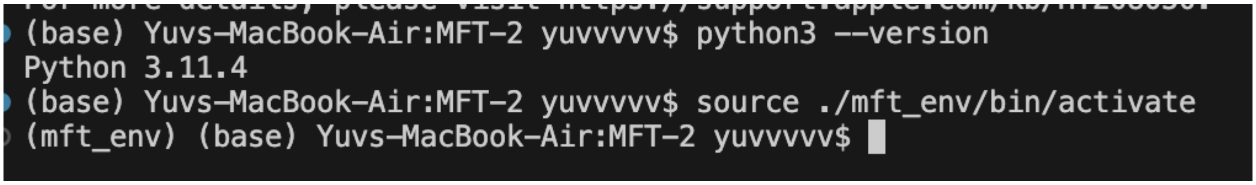
Research Documentation for Data Collation on South East Asian Non-Profits

By: Yuv Bindal

Table of Contents:

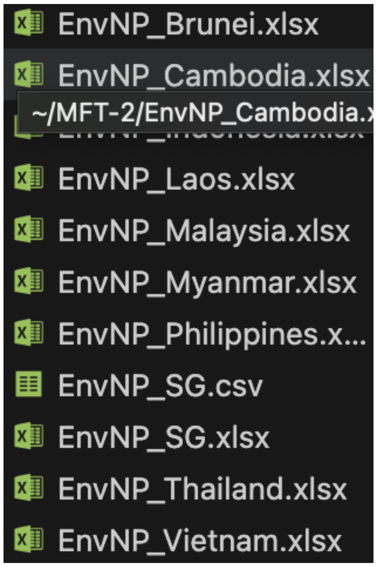
Setup

1. Clone the repository: https://github.com/YuvBindal/MFT (Only available to granted collaborators)
2. Activate the virtual environment '**mft_env**' to install the dependencies.



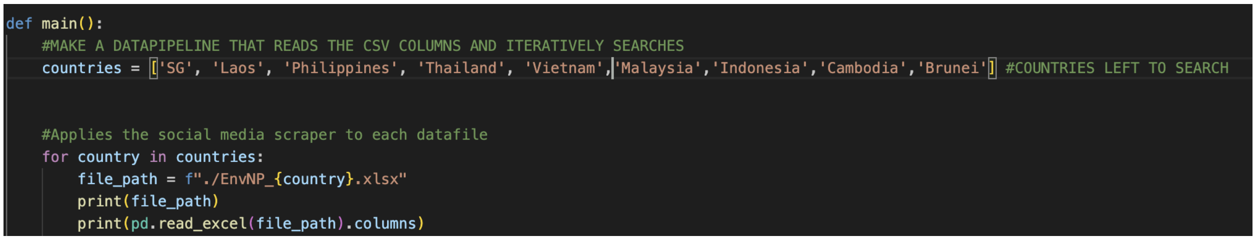
```
For more details, please visit https://support.apple.com/kb/HT208050.
(base) Yuvs-MacBook-Air:MFT-2 yuvvvvv$ python3 --version
Python 3.11.4
(base) Yuvs-MacBook-Air:MFT-2 yuvvvvv$ source ./mft_env/bin/activate
(mft_env) (base) Yuvs-MacBook-Air:MFT-2 yuvvvvv$
```

3. Prepare Individual excel/csv data files for each respective country in the name format "EnvNP_{country_name_here}.xlsx"



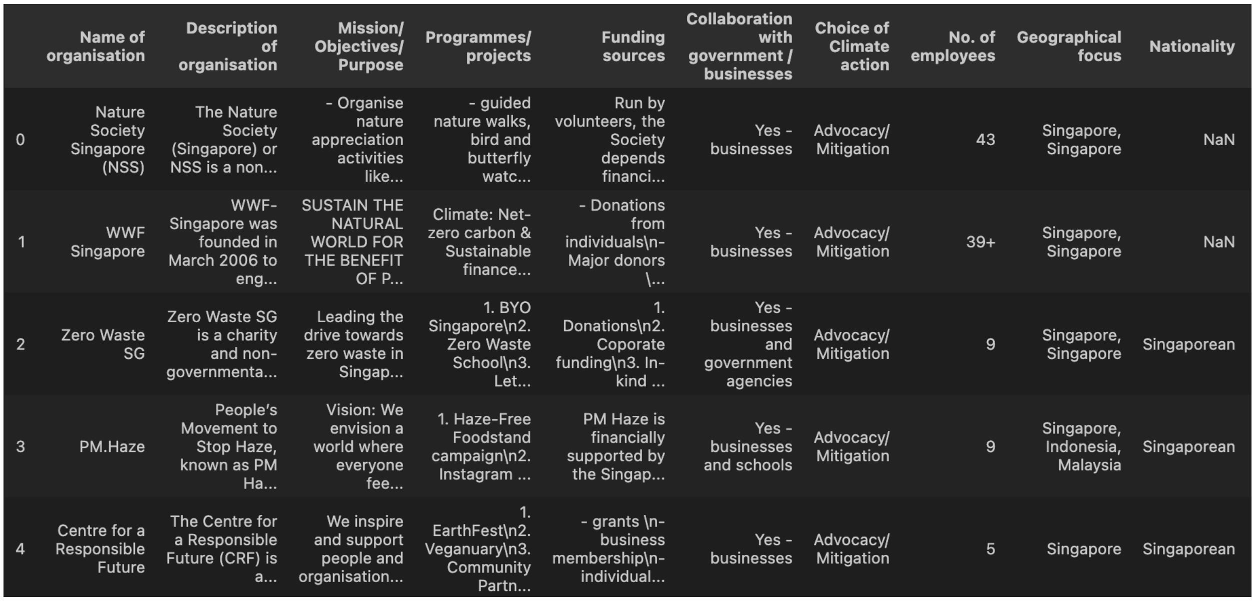
```
EnvNP_Brunei.xlsx
EnvNP_Cambodia.xlsx
~/MFT-2/EnvNP_Cambodia.x
EnvNP_Laos.xlsx
EnvNP_Malaysia.xlsx
EnvNP_Myanmar.xlsx
EnvNP_Philippines.x...
EnvNP_SG.csv
EnvNP_SG.xlsx
EnvNP_Thailand.xlsx
EnvNP_Vietnam.xlsx
```

4. Open SocialScrapBots.py and run the cell to import all the files using the following loop:



```
def main():
    #MAKE A DATAPIPELINE THAT READS THE CSV COLUMNS AND ITERATIVELY SEARCHES
    countries = ['SG', 'Laos', 'Philippines', 'Thailand', 'Vietnam','Malaysia','Indonesia','Cambodia','Brunei'] #COUNTRIES LEFT TO SEARCH


    #Applies the social media scraper to each datafile
    for country in countries:
        file_path = f"./EnvNP_{country}.xlsx"
        print(file_path)
        print(pd.read_excel(file_path).columns)
```

5. Evaluate the structure of the files to ensure they are imported correctly as follows:

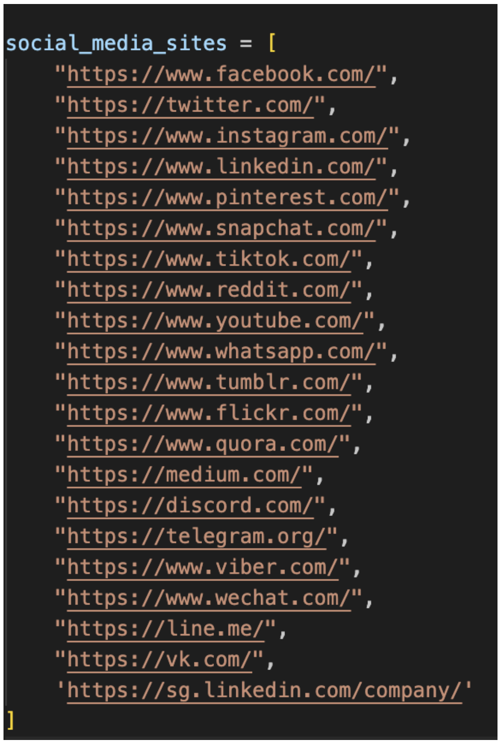


	Name of organisation	Description of organisation	Mission/ Objectives/ Purpose	Programmes/ projects	Funding sources	Collaboration with government / businesses	Choice of Climate action	No. of employees	Geographical focus	Nationality
0	Nature Society Singapore (NSS)	The Nature Society (Singapore) or NSS is a non...	- Organise nature appreciation activities like...	- guided nature walks, bird and butterfly watc...	Run by volunteers, the Society depends financi...	Yes - businesses	Advocacy/ Mitigation	43	Singapore, Singapore	NaN
1	WWF Singapore	WWF-Singapore was founded in March 2006 to eng...	SUSTAIN THE NATURAL WORLD FOR THE BENEFIT OF P...	Climate: Net-zero carbon & Sustainable finance...	- Donations from individuals\n- Major donors \...	Yes - businesses	Advocacy/ Mitigation	39+	Singapore, Singapore	NaN
2	Zero Waste SG	Zero Waste SG is a charity and non-governmenta...	Leading the drive towards zero waste in Singap...	1. BYO Singapore\n2. Zero Waste School\n3. Let...	1. Donations\n2. Coporate funding\n3. In-kind ...	Yes - businesses and government agencies	Advocacy/ Mitigation	9	Singapore, Singapore	Singaporean
3	PM.Haze	People's Movement to Stop Haze, known as PM Ha...	Vision: We envision a world where everyone fee...	1. Haze-Free Foodstand campaign\n2. Instagram ...	PM Haze is financially supported by the Singap...	Yes - businesses and schools	Advocacy/ Mitigation	9	Singapore, Indonesia, Malaysia	Singaporean
4	Centre for a Responsible Future	The Centre for a Responsible Future (CRF) is a...	We inspire and support people and organisation...	1. EarthFest\n2. Veganuary\n3. Community Partn...	- grants \n- business membership\n- individual...	Yes - businesses	Advocacy/ Mitigation	5	Singapore	Singaporean

Web Scraping for Information

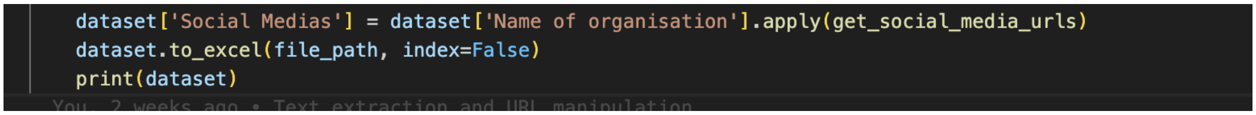
Part 1: Social Media Links

1. Prepare a list of social media sites to extract from search queries from the net:



```
social_media_sites = [
    "https://www.facebook.com/",
    "https://twitter.com/",
    "https://www.instagram.com/",
    "https://www.linkedin.com/",
    "https://www.pinterest.com/",
    "https://www.snapchat.com/",
    "https://www.tiktok.com/",
    "https://www.reddit.com/",
    "https://www.youtube.com/",
    "https://www.whatsapp.com/",
    "https://www.tumblr.com/",
    "https://www.flickr.com/",
    "https://www.quora.com/",
    "https://medium.com/",
    "https://discord.com/",
    "https://telegram.org/",
    "https://www.viber.com/",
    "https://www.wechat.com/",
    "https://line.me/",
    "https://vk.com/",
    'https://sg.linkedin.com/company/'
]
```

2. Apply a get_social_media_urls to define a new feature column “Social Medias” in the dataset. It should be noted dataset.to_excel() saves the dataset locally as a xlsx file after applying the function.



```
dataset['Social Medias'] = dataset['Name of organisation'].apply(get_social_media_urls)
dataset.to_excel(file_path, index=False)
print(dataset)
```

3. Define a get_social_media_urls function to extract social media urls for each company

   Overview:

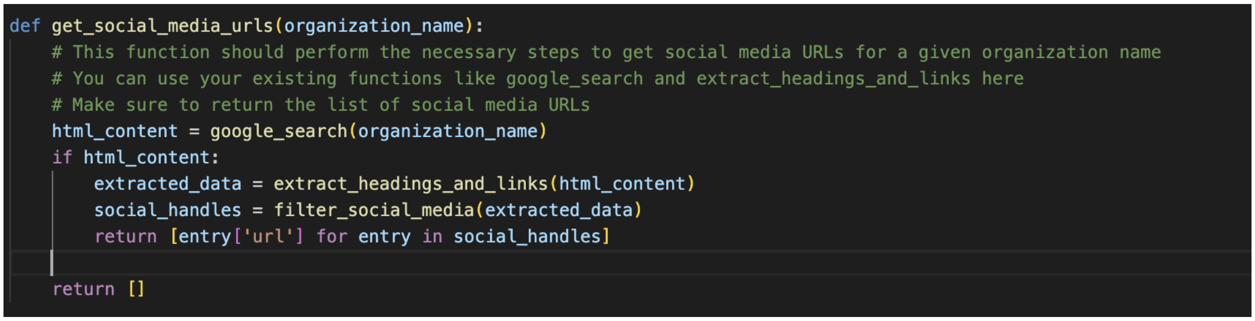
   - Organisation_name is fed as a parameter to a function google_search.

   - The google_search function makes a search query on the net and retrieves the html_content for that query.

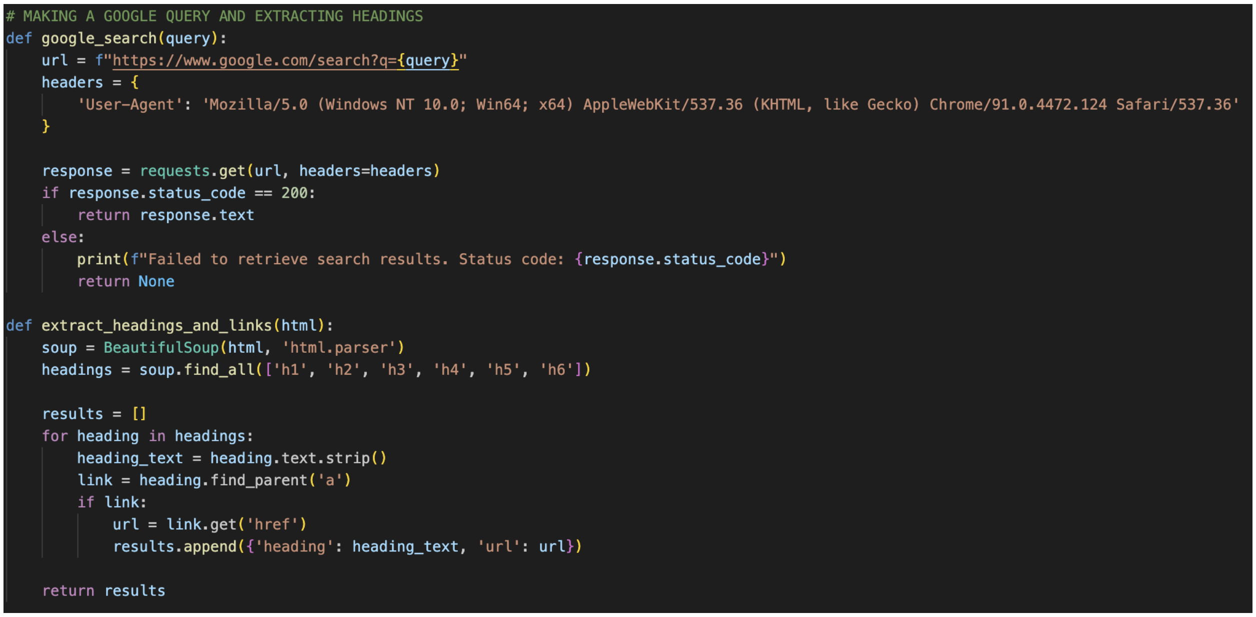
   - If this html_content is successfully retrieved, then we call the extract_headings_and_links() function that extracts all links and titles of those links from the html_content.

   - These links are fed into the filter_social_media() function that finally filters only the list containing headers and links to social media sites.

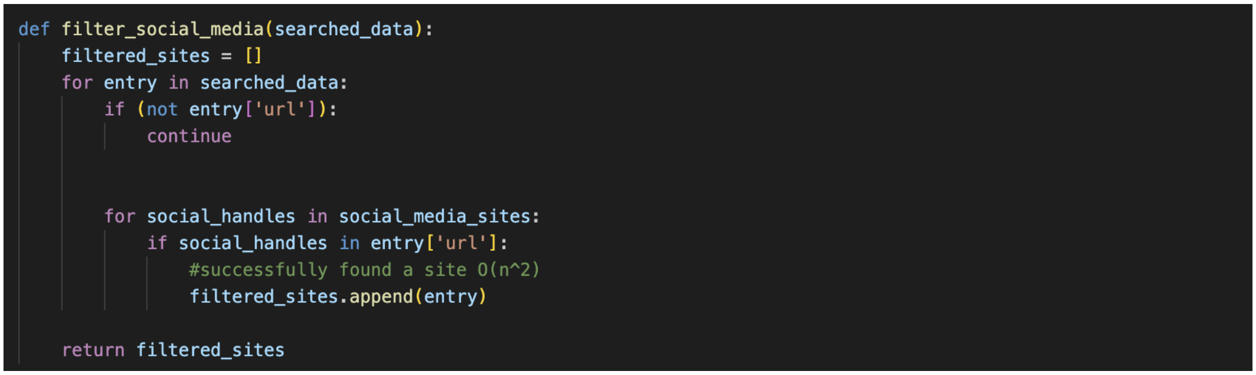
- Finally, a list comprehension is used to extract the links for these social media sites based on the previously described list.



```python
def get_social_media_urls(organization_name):
    # This function should perform the necessary steps to get social media URLs for a given organization name
    # You can use your existing functions like google_search and extract_headings_and_links here
    # Make sure to return the list of social media URLs
    html_content = google_search(organization_name)
    if html_content:
        extracted_data = extract_headings_and_links(html_content)
        social_handles = filter_social_media(extracted_data)
        return [entry['url'] for entry in social_handles]

    return []
```

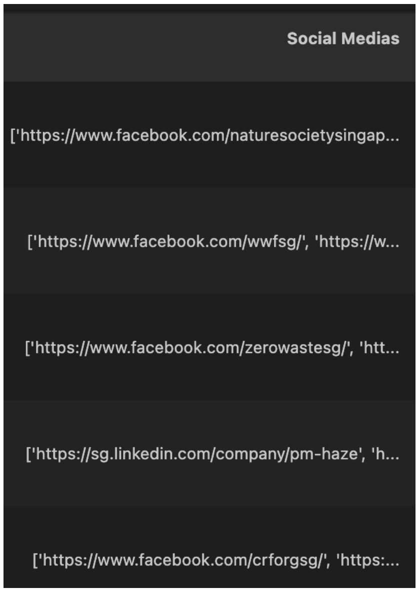


```python
# MAKING A GOOGLE QUERY AND EXTRACTING HEADINGS
def google_search(query):
    url = f"https://www.google.com/search?q={query}"
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.124 Safari/537.36'
    }

    response = requests.get(url, headers=headers)
    if response.status_code == 200:
        return response.text
    else:
        print(f"Failed to retrieve search results. Status code: {response.status_code}")
        return None

def extract_headings_and_links(html):
    soup = BeautifulSoup(html, 'html.parser')
    headings = soup.find_all(['h1', 'h2', 'h3', 'h4', 'h5', 'h6'])

    results = []
    for heading in headings:
        heading_text = heading.text.strip()
        link = heading.find_parent('a')
        if link:
            url = link.get('href')
            results.append({'heading': heading_text, 'url': url})

    return results
```



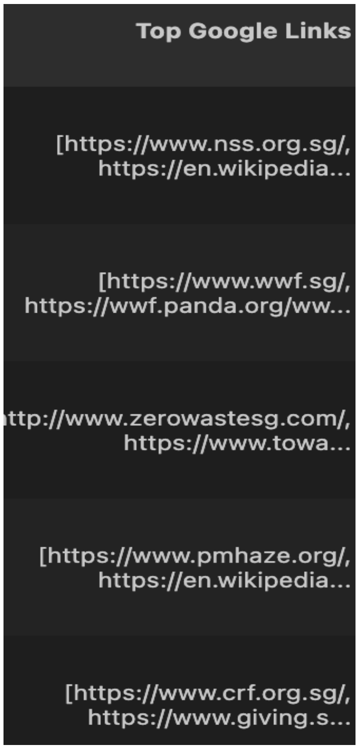
```python
def filter_social_media(searched_data):
    filtered_sites = []
    for entry in searched_data:
        if (not entry['url']):
            continue


        for social_handles in social_media_sites:
            if social_handles in entry['url']:
                #successfully found a site O(n^2)
                filtered_sites.append(entry)

    return filtered_sites
```

4. The apply() function might take time to create the new feature column as web scraping is a computational expensive process applied to over 200 observations collated from all datasets. Finally, the resultant column should look as follows:

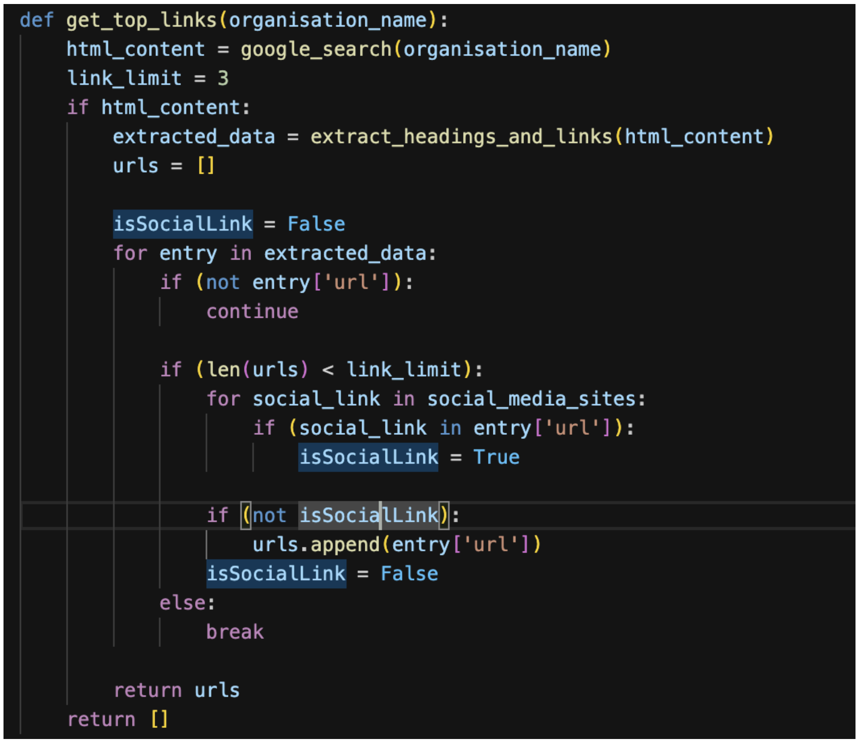


Social Medias
['https://www.facebook.com/naturesocietysingap...
['https://www.facebook.com/wwfsg/', 'https://w...
['https://www.facebook.com/zerowastesg/', 'htt...
['https://sg.linkedin.com/company/pm-haze', 'h...
['https://www.facebook.com/crforgsg/', 'https:...

5. Similarly, a function can be developed to extract all other links that are not social media links as follows:



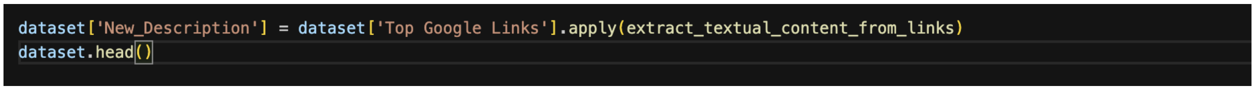
Top Google Links
[https://www.nss.org.sg/, https://en.wikipedia...
[https://www.wwf.sg/, https://wwf.panda.org/ww...
ttp://www.zerowastesg.com/, https://www.towa...
[https://www.pmhaze.org/, https://en.wikipedia...
[https://www.crf.org.sg/, https://www.giving.s...



```
def get_top_links(organisation_name):
    html_content = google_search(organisation_name)
    link_limit = 3
    if html_content:
        extracted_data = extract_headings_and_links(html_content)
        urls = []

        isSocialLink = False
        for entry in extracted_data:
            if (not entry['url']):
                continue

            if (len(urls) < link_limit):
                for social_link in social_media_sites:
                    if (social_link in entry['url']):
                        isSocialLink = True

                if (not isSocialLink):
                    urls.append(entry['url'])
                isSocialLink = False
            else:
                break

        return urls
    return []
```

Part 2: Extracting Textual Content from Links

1. Define a new feature column 'New Description' to store textual content extracted



```
dataset['New_Description'] = dataset['Top Google Links'].apply(extract_textual_content_from_links)
dataset.head()
```

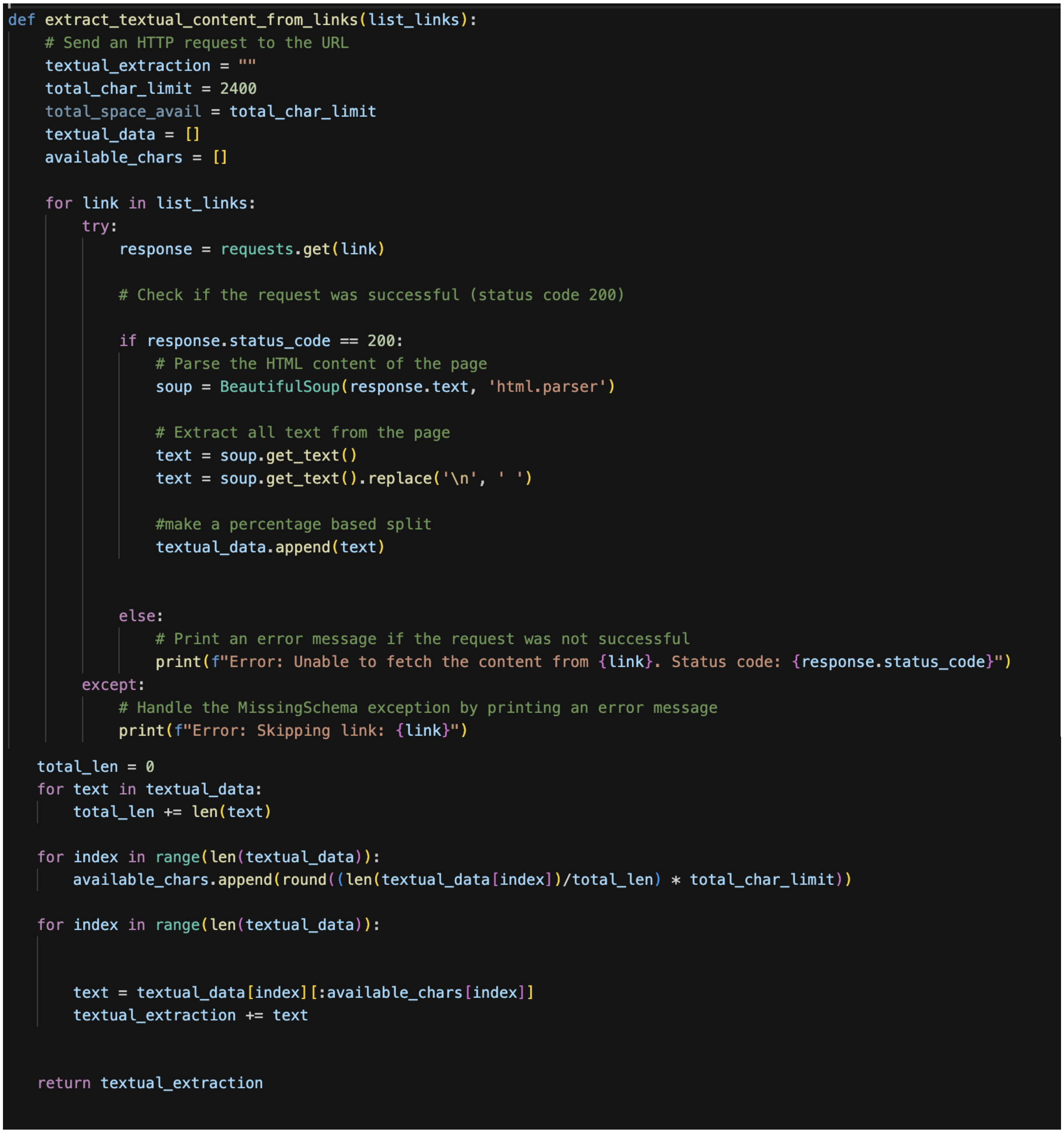
2. Define a function to extract textual content from links as follows:

   Overview:

   - Set a character limit such that textual content extracted does not exceed memory considerations

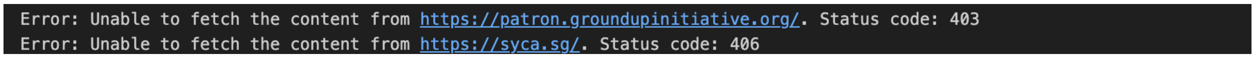
   - Loop through the links and store the total textual content extracted from the main page in a list, removing newline characters.

   - Lastly, a percentage split based approach is taken when appending the final extracted string. For example, given 3 links if link 1 extracts 1000 characters, link 2 extracts 1500 characters, and link 3 extracts 500 characters. Then, total length extracted = 1000 + 1500 + 500 = 3000 characters. However, the limit is 2400. To compensate this we take percentages to assign weights to each link in the following manner: link 1: 33.3% (1000/3000), link 2: 50% (1500/3000), link 3: 16.7% (500/3000). Thus total characters appended to final string from respective links are: **link 1: 800 (.333 *2400), link 2: 1200 (.5 * 2400), link 3: 400 (.167*2400)**



```python
def extract_textual_content_from_links(list_links):
    # Send an HTTP request to the URL
    textual_extraction = ""
    total_char_limit = 2400
    total_space_avail = total_char_limit
    textual_data = []
    available_chars = []

    for link in list_links:
        try:
            response = requests.get(link)

            # Check if the request was successful (status code 200)

            if response.status_code == 200:
                # Parse the HTML content of the page
                soup = BeautifulSoup(response.text, 'html.parser')

                # Extract all text from the page
                text = soup.get_text()
                text = soup.get_text().replace('\n', ' ')

                #make a percentage based split
                textual_data.append(text)


            else:
                # Print an error message if the request was not successful
                print(f"Error: Unable to fetch the content from {link}. Status code: {response.status_code}")
        except:
            # Handle the MissingSchema exception by printing an error message
            print(f"Error: Skipping link: {link}")

   total_len = 0
   for text in textual_data:
       total_len += len(text)

   for index in range(len(textual_data)):
       available_chars.append(round((len(textual_data[index])/total_len) * total_char_limit))

   for index in range(len(textual_data)):


       text = textual_data[index][:available_chars[index]]
       textual_extraction += text


   return textual_extraction
```

It should be noted that while scraping textual content from certain websites, the following API error can be thrown. These are normal as certain websites block web scrapers.

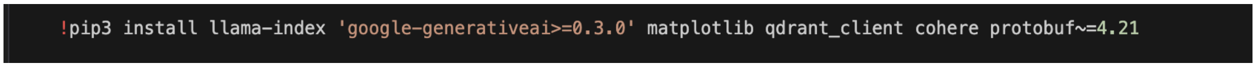


```
Error: Unable to fetch the content from https://patron.groundupinitiative.org/. Status code: 403
Error: Unable to fetch the content from https://syca.sg/. Status code: 406
```

Part 3: Use Generative AI to extract information from scraped datasets

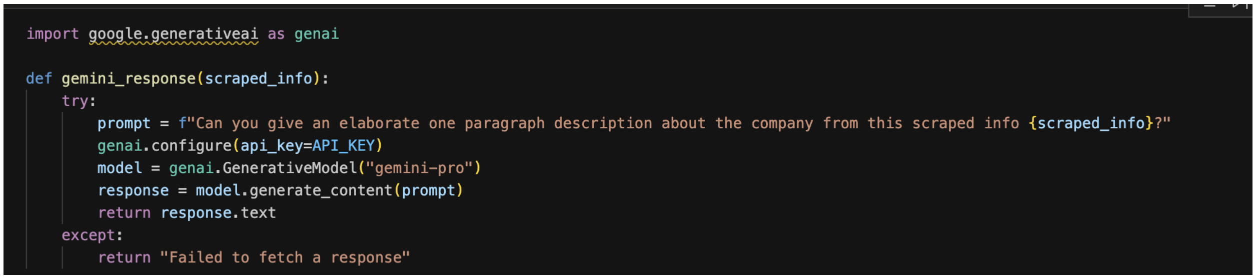
This research uses Google's Gemini Large Language Model to elaborate on the scraped company information from the websites. To use this API, you must obtain a Google Cloud API key. For more information, you can visit: https://cloud.google.com/.

1. The virtual environment mft_env comes with the necessary dependencies for Google's generative AI models. However, they can be downloaded in Python3 using the following commands:



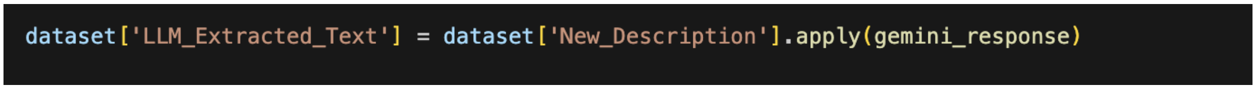
```
!pip3 install llama-index 'google-generativeai>=0.3.0' matplotlib qdrant_client cohere protobuf~=4.21
```

2. Define a function to interact with the AI model. Namely, define a prompt that commands the model what it needs to achieve; as well as, select the appropriate version of the Gemini model. In this case, we are using "Gemini-pro" as we just require textual information. For multimodal purposes, "Gemini-pro-vision" can be used.



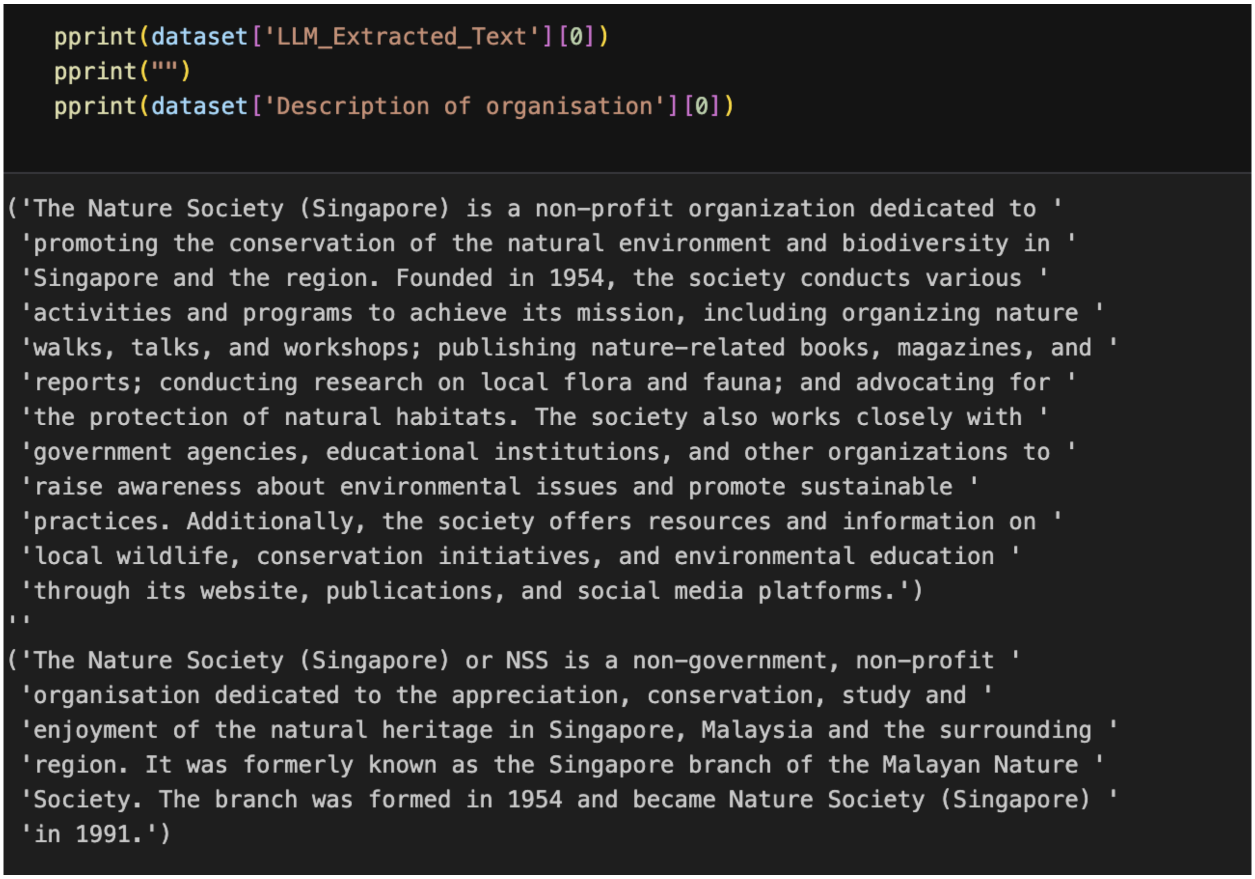
```
import google.generativeai as genai

def gemini_response(scraped_info):
    try:
        prompt = f"Can you give an elaborate one paragraph description about the company from this scraped info {scraped_info}?"
        genai.configure(api_key=API_KEY)
        model = genai.GenerativeModel("gemini-pro")
        response = model.generate_content(prompt)
        return response.text
    except:
        return "Failed to fetch a response"
```

3. Define a new feature column to store LLM Responses and apply the following function to the dataset as follows:



```
dataset['LLM_Extracted_Text'] = dataset['New_Description'].apply(gemini_response)
```

4. Lastly, evaluate the retrieved description quality with the previously manually inputted values. A significant improvement in description quality is noticed; as well as, the process is automated. Disclaimer, this process usually works well with bigger companies that have largely available information online.



```
   pprint(dataset['LLM_Extracted_Text'][0])
   pprint("")
   pprint(dataset['Description of organisation'][0])
```

```
('The Nature Society (Singapore) is a non-profit organization dedicated to '
 'promoting the conservation of the natural environment and biodiversity in '
 'Singapore and the region. Founded in 1954, the society conducts various '
 'activities and programs to achieve its mission, including organizing nature '
 'walks, talks, and workshops; publishing nature-related books, magazines, and '
 'reports; conducting research on local flora and fauna; and advocating for '
 'the protection of natural habitats. The society also works closely with '
 'government agencies, educational institutions, and other organizations to '
 'raise awareness about environmental issues and promote sustainable '
 'practices. Additionally, the society offers resources and information on '
 'local wildlife, conservation initiatives, and environmental education '
 'through its website, publications, and social media platforms.')
''
('The Nature Society (Singapore) or NSS is a non-government, non-profit '
 'organisation dedicated to the appreciation, conservation, study and '
 'enjoyment of the natural heritage in Singapore, Malaysia and the surrounding '
 'region. It was formerly known as the Singapore branch of the Malayan Nature '
 'Society. The branch was formed in 1954 and became Nature Society (Singapore) '
 'in 1991.')
```

Part 4: Social Media Analysis

As of February 8th, 2024, this is an going area of research and the document will be updated once significant data is retrieved.